

Mirek,

Please check out this one. It features better performance on MSBT19 / MSBT19x64 with low counts, and works well on CLANG/CLANGx64 too:

```
inline void new_memset128(void *b, dword data, int len){
```

```
    switch(len){
    case 4: ((dword *)b)[3] = data;
    case 3: ((dword *)b)[2] = data;
    case 2: ((dword *)b)[1] = data;
    case 1: ((dword *)b)[0] = data;
    case 0: return;
    }
```

```
    __m128i q = _mm_set1_epi32(*(int*)&data);
    __m128i *w = (__m128i*)b;
```

```
    switch(len>>2){
    default:{
        __m128i *e = (__m128i*)b + (len>>2) - 4;
        if(len <= 2*1024*1024){
            while(w<e){
                _mm_storeu_si128(w++, q);
                _mm_storeu_si128(w++, q);
                _mm_storeu_si128(w++, q);
                _mm_storeu_si128(w++, q);
            }
        }
        else{
            while(w<e){
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
                _mm_stream_si128(w++, q);
            }
        }
    }
    case 4: _mm_storeu_si128(w++, q);
    case 3: _mm_storeu_si128(w++, q);
    case 2: _mm_storeu_si128(w++, q);
    case 1: _mm_storeu_si128(w++, q);
    }
    switch(len&3){
    case 3: ((dword *)b)[len-3] = data;
```

```
case 2: ((dword *)b)[len-2] = data;  
case 1: ((dword *)b)[len-1] = data;  
}  
}
```

Best regards,

Tom

EDIT: Fine tuning...
